

中華民國經濟部智慧財產局

INTELLECTUAL PROPERTY OFFICE  
MINISTRY OF ECONOMIC AFFAIRS  
REPUBLIC OF CHINA

茲證明所附文件，係本局存檔中原申請案的副本，正確無訛，  
其申請資料如下：

This is to certify that annexed is a true copy from the records of this  
office of the application as originally filed which is identified hereunder:

申請日：西元 2003 年 05 月 19 日  
Application Date

申請案號：092113483  
Application No.

申請人：財團法人工業技術研究院  
Applicant(s)

局長  
Director-General

蔡練生

發文日期：西元 2003 年 8 月 8 日  
Issue Date

發文字號：09220801670  
Serial No.

# 發明專利說明書

(填寫本書件時請先行詳閱申請書後之申請須知，作※記號部分請勿填寫)

※ 申請案號：\_\_\_\_\_ ※IPC 分類：\_\_\_\_\_

※ 申請日期：\_\_\_\_\_

## 壹、發明名稱

(中文) 轉換函數之硬體架構設計方法及裝置

(英文) Method and apparatus of constructing a hardware architecture for transfer functions

## 貳、發明人 (共 3 人)

發明人 1 (如發明人超過一人，請填說明書發明人續頁)

姓名：(中文) 陳信宏

(英文) Hsin-hung Chen

住居所地址：(中文) 高雄縣鳳山市大德里公園街 375 巷 20 號

(英文) No. 20, Lane 375, Gungyuan 1st St., Fengshan City, Kaohsiung

國籍：(中文) 中華民國

(英文) R.O.C.

## 參、申請人 (共 1 人)

申請人 1 (如發明人超過一人，請填說明書申請人續頁)

姓名或名稱：(中文) 財團法人工業技術研究院

(英文) Industrial Technology Research Institute

住居所或營業所地址：(中文) 新竹縣竹東鎮中興路四段一九五號

(英文) No. 195, Sec. 4, Chung Hsing Rd., Chutung, Hsinchu

國籍：(中文) 中華民國

(英文) R.O.C.

代表人：(中文) 翁政義

(英文) Cheng-I Weng

☐ 續發明人或申請人續頁 (發明人或申請人欄位不敷使用時，請註記並使用續頁)

發明人 2

姓名：(中文) 陳自強

(英文) Oscal T.-C. Chen

住居所地址：(中文) 台北縣永和市得和路 330 號 4 樓

(英文) 4Fl., No. 330, Dehe Rd., Yunghe City, Taipei

國籍：(中文) 中華民國

(英文) R.O.C.

發明人 3

姓名：(中文) 葉恆誠

(英文) Roger, Heng-cheng Yeh

住居所地址：(中文) 台北市士林區雨農路 2 巷 17 號 3 樓

(英文) 3Fl., No. 17, Lane 2, Yunung Rd., Shrlin Chiu, Taipei

國籍：(中文) 中華民國

(英文) R.O.C.

#### 肆、中文發明摘要

本發明係有關於一種轉換函數之硬體架構設計方法及其裝置，其係採用單點輸入、平行輸出的方式來進行運算，並將轉換函數中之運算分為乘法運算、路徑選擇、及累加運算。首先，將輸入訊號乘以所有轉換係數的數值；接著再利用路徑選擇器選擇正確乘積結果訊號傳送給對應時序之累加器進行運算；最後再將累加結果乘以常數項以計算出輸出訊號。本發明係利用轉換係數間之對稱性，以減少乘法運算個數來降低硬體架構複雜度；並使用加減法器取代固定數值乘法器，且利用共享子項來減少加減法器之個數以精簡架構。

#### 伍、英文發明摘要

A method and apparatus of constructing a hardware architecture for transfer functions is disclosed, which uses a single-input-parallel-output algorithm for processing operations. The transfer function has operations of multiplication, path-selection, and accumulation to be executed. The fixed-value multipliers first multiply an input signal by all transfer coefficients. Then a path-selection unit determines correct signal paths and delivers product results to the corresponding accumulators for processing accumulation. Finally, multipliers perform the multiplications of the accumulated values and a constant to obtain output signals. The present invention can reduce the number of processing multiplications according to the symmetry among transfer coefficients, use adders to replace fixed-value multipliers, and reduce the number of adders by utilizing shared terms to simplify the hardware architecture.

陸、(一)、本案指定代表圖為：圖 2

(二)、本代表圖之元件代表符號簡單說明：

輸入單元 11

固定數值乘法器 12

路徑選擇器 13

控制器 131

累加器 141,142,143

乘法器 151,152,153

輸出單元 16

柒、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

無

## 捌、聲明事項

☐ 本案係符合專利法第二十條第一項第一款但書或第二款但書規定之期間，其日期為：\_\_\_\_\_

☐ 本案已向下列國家（地區）申請專利，申請日期及案號資料如下：

【格式請依：申請國家（地區）；申請日期；申請案號 順序註記】

1. 無

2. \_\_\_\_\_

3. \_\_\_\_\_

☐ 主張專利法第二十四條第一項優先權：

【格式請依：受理國家（地區）；日期；案號 順序註記】

1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

4. \_\_\_\_\_

5. \_\_\_\_\_

6. \_\_\_\_\_

7. \_\_\_\_\_

8. \_\_\_\_\_

9. \_\_\_\_\_

10. \_\_\_\_\_

☐ 主張專利法第二十五條之一第一項優先權：

【格式請依：申請日；申請案號 順序註記】

1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

☐ 主張專利法第三十六條微生物：

☐ 國內微生物 【格式請依：寄存機構；日期；號碼 順序註記】

1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

☐ 國外微生物 【格式請依：寄存國名；機構；日期；號碼 順序註記】

1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

☐ 熟習該項技術者易於獲得，不須寄存。

## 發明說明

(發明說明應敘明：發明所屬之技術領域、先前技術、內容、實施方式及圖式簡單說明)

### 一、發明所屬之技術領域

本發明係關於一種硬體架構設計方法，尤指一種轉換函數之硬體架構設計方法及裝置，其適用範圍包括應用於以相乘和累加方式實現、且具有固定係數架構之轉換函數電路中。

### 二、先前技術

在數位訊號處理技術領域中，經常使用轉換函數以藉由訊號的物理特性來進行領域間的信號轉換，例如時域與頻域間的轉換，俾便後續的訊號處理程序。

一般而言，轉換函數需要處理許多相乘與累加的運算，例如如下式所示之四點離散傅立葉轉換 (Discrete Fourier Transform, DFT)：

$$\begin{aligned} y(k) &= \sum_{n=0}^3 x(n) e^{-j \frac{2nk\pi}{4}} \\ y(0) &= x(0) e^{-j \frac{0\pi}{4}} + x(1) e^{-j \frac{0\pi}{4}} + x(2) e^{-j \frac{0\pi}{4}} + x(3) e^{-j \frac{0\pi}{4}} \\ y(1) &= x(0) e^{-j \frac{0\pi}{4}} + x(1) e^{-j \frac{2\pi}{4}} + x(2) e^{-j \frac{4\pi}{4}} + x(3) e^{-j \frac{6\pi}{4}}, \\ y(2) &= x(0) e^{-j \frac{0\pi}{4}} + x(1) e^{-j \frac{4\pi}{4}} + x(2) e^{-j \frac{8\pi}{4}} + x(3) e^{-j \frac{12\pi}{4}} \\ y(3) &= x(0) e^{-j \frac{0\pi}{4}} + x(1) e^{-j \frac{6\pi}{4}} + x(2) e^{-j \frac{12\pi}{4}} + x(3) e^{-j \frac{18\pi}{4}} \end{aligned}$$

其中， $y(k)$ 為轉換後之輸出訊號， $x(n)$ 為輸入訊號。習知在進行上述離散傅立葉轉換過程中，大多係採用平行化處理，亦即使用多套相乘累加器來對 $y(0)$ 、 $y(1)$ 、 $y(2)$ 及 $y(3)$ 分別做乘法及累加的運算；或可僅採用一套相乘累加器，藉由重複的使用來達到節省硬體架構與面積之功能；另亦

有利用轉換函數的特性來設計出制式硬體架構或演算法，例如快速傅立葉轉換(Fast Fourier Transform, FFT)即為利用離散傅立葉轉換之特性所推演出來的結果。

前述四點離散傅立葉轉換公式可以矩陣表示為：

$$5 \quad \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = T x = \begin{bmatrix} e^{-j\frac{0\pi}{4}} & e^{-j\frac{0\pi}{4}} & e^{-j\frac{0\pi}{4}} & e^{-j\frac{0\pi}{4}} \\ e^{-j\frac{0\pi}{4}} & e^{-j\frac{2\pi}{4}} & e^{-j\frac{4\pi}{4}} & e^{-j\frac{6\pi}{4}} \\ e^{-j\frac{0\pi}{4}} & e^{-j\frac{4\pi}{4}} & e^{-j\frac{8\pi}{4}} & e^{-j\frac{12\pi}{4}} \\ e^{-j\frac{0\pi}{4}} & e^{-j\frac{6\pi}{4}} & e^{-j\frac{12\pi}{4}} & e^{-j\frac{18\pi}{4}} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix},$$

其中，T為轉換矩陣，且矩陣內之係數為轉換係數。由於在前述矩陣中，有部分轉換係數值其實是相同的，因此可將具有相同數值之轉換係數根據下列公式加以簡化：

$$e^{j(\theta+2l\pi)} = e^{j\theta} \quad l \in \text{整數},$$

10 進而形成如下所示之矩陣：

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} e^{-j\frac{0\pi}{4}} & e^{-j\frac{0\pi}{4}} & e^{-j\frac{0\pi}{4}} & e^{-j\frac{0\pi}{4}} \\ e^{-j\frac{0\pi}{4}} & e^{-j\frac{2\pi}{4}} & e^{-j\frac{4\pi}{4}} & e^{-j\frac{6\pi}{4}} \\ e^{-j\frac{0\pi}{4}} & e^{-j\frac{4\pi}{4}} & e^{-j\frac{8\pi}{4}} & e^{-j\frac{12\pi}{4}} \\ e^{-j\frac{0\pi}{4}} & e^{-j\frac{6\pi}{4}} & e^{-j\frac{12\pi}{4}} & e^{-j\frac{18\pi}{4}} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix},$$

亦即  $e^{-j\frac{8\pi}{4}} = e^{-j\frac{0\pi}{4}}$ 、 $e^{-j\frac{12\pi}{4}} = e^{-j\frac{4\pi}{4}}$ 、且  $e^{-j\frac{18\pi}{4}} = e^{-j\frac{2\pi}{4}}$ ...餘此類推。

然而，由於傳統轉換函數的設計是將輸入訊號以時序的方式輸入，如下列公式所示：



$$\begin{aligned}
y(0) &= \boxed{x(0)e^{-j\frac{0\pi}{4}}} + \boxed{x(1)e^{-j\frac{0\pi}{4}}} + \boxed{x(2)e^{-j\frac{0\pi}{4}}} + \boxed{x(3)e^{-j\frac{0\pi}{4}}} \\
y(1) &= \boxed{x(0)e^{-j\frac{0\pi}{4}}} + \boxed{x(1)e^{-j\frac{2\pi}{4}}} + \boxed{x(2)e^{-j\frac{4\pi}{4}}} + \boxed{x(3)e^{-j\frac{6\pi}{4}}} \\
y(2) &= \boxed{x(0)e^{-j\frac{0\pi}{4}}} + \boxed{x(1)e^{-j\frac{4\pi}{4}}} + \boxed{x(2)e^{-j\frac{0\pi}{4}}} + \boxed{x(3)e^{-j\frac{4\pi}{4}}} \\
y(3) &= \boxed{x(0)e^{-j\frac{0\pi}{4}}} + \boxed{x(1)e^{-j\frac{6\pi}{4}}} + \boxed{x(2)e^{-j\frac{4\pi}{4}}} + \boxed{x(3)e^{-j\frac{2\pi}{4}}}
\end{aligned}$$

其中，以虛線框起來的部分分別代表每一個時序(n=0、1、2、3)上所要處理的乘法運算，請一併參閱圖1，傳統的作法大多係利用四套相乘累加器平行處理來實現硬體架構，而其中之  $T_c(k,n)$  係為轉換矩陣中第k行、第n列的轉換係數值。

然而，雖然可得知k值，但n還是會隨著輸入訊號的時序而改變，並非固定不變的數值，因此必須利用記憶體來儲存係數值，之後再依據時序讀入對應係數來進行乘法運算。根據上述說明，顯示習知利用多個乘法器分時多工，並在不同時間載入對應係數與輸入訊號來進行相乘累加運算後產生輸出訊號的方法，由於乘法器所耗用之系統資源極為龐大，因此不但使硬體架構設計之複雜度提高，還會加重系統運算負荷量導致運算效率不佳。由此可知，習知轉換函數之硬體架構設計方法仍存在有諸多缺失而有予以改進之必要。

### 三、發明內容

本發明之主要目的係在提供一種轉換函數之硬體架構設計方法及裝置，係使用加法器及減法器以取代乘法器來實現固定係數之乘法運算，俾能減少系統運算負荷。

5 本發明之另一目的係在提供一種轉換函數之硬體架構設計方法及裝置，係利用共享子項(shared term)結合相同轉換係數，俾以減少加法器及減法器之個數以精簡硬體架構，並能降低硬體成本、提高運算效率，同時達成轉換函數要求的精確度。

10 依據本發明之一特色，所提出之轉換函數之硬體架構設計方法包括下述步驟，首先係擷取一轉換函數，其用以將一特定領域之輸入訊號 $x(n)$ 轉換為另一特定領域之輸出訊號 $y(k)$ ；接著將轉換函數中具有相同數值之轉換係數統一為一致之轉換係數，其中，每一特定數值之轉換係數係對應定義有一固定數值乘法器；之後分別使用固定數值乘法器將輸入訊號與對應之轉換係數進行相乘運算；再使用  
15 一路徑選擇器以根據轉換函數之定義而將輸入訊號與對應轉換函數之相乘結果分配至輸出訊號之時序所對應之累加器；在使用累加器將位於對應時序之相乘結果進行累加運算；最後使用乘法器將累加結果乘以轉換函數之常數項以  
20 計算出輸出訊號後，將此輸出訊號加以輸出。

依據本發明之另一特色，係提出一種轉換函數之硬體裝置，其主要包括一輸入單元、至少一固定數值乘法器、至少一路徑選擇器、至少一累加器、及一輸出單元。此轉換函數是將一特定領域之輸入訊號 $x(n)$ 轉換為另一特定領

域之輸出訊號  $y(k)$ ；其中輸入單元係用以接收輸入訊號，固定數值乘法器係用以將輸入訊號與轉換函數中所定義之對應轉換係數進行相乘運算；路徑選擇器可根據轉換函數之定義以將輸入訊號與對應轉換函數之相乘結果分配至輸出訊號之時序所對應之累加器；每一累加器則對應於一特定時序，用以將位於其時序之相乘結果進行累加運算；並透過乘法器將累加結果乘以轉換函數之常數項以計算出輸出訊號；最後由輸出單元將輸出訊號加以輸出。

#### 10 四、實施方式

為能讓貴審查委員能更瞭解本發明之技術內容，特舉二具體實施例說明如下。

本發明所提出之轉換函數之硬體架構設計方法及裝置可適用任何如下列方程式所表示之轉換函數中：

$$15 \quad y(k) = A \sum_{n=0}^{N-1} T_c(k,n)x(n) \quad k=0,1,2,\dots,N-1,$$

其中， $x(n)$ 係為一特定領域(domain)之輸入訊號， $y(k)$ 為另一特定領域之輸出訊號，且 $A$ 為常數項， $T_c(k,n)$ 為轉換係數值，並將隨著不同的輸出項及輸入項而有所改變。其中，當轉換函數係用以進行反向離散傅立葉轉換(Inverse

20 Discrete Fourier Transform, IDFT)時， $A$ 等於 $\frac{1}{N}$ ；當然轉換

函數亦可用於進行離散傅立葉轉換(DFT)、離散餘弦轉換(DCT)/反向離散餘弦轉換(IDCT)、及離散正弦轉換(DST)/

反向離散正弦轉換(IDST)等，且較佳係應用於進行單點輸入及平行輸出(single input parallel output)之環境中。

在進行本發明轉換函數之硬體架構設計時，可先將前述方程式展開為：

$$\begin{aligned} y(0) &= A \sum_{n=0}^{N-1} T_c(0,n)x(n) \\ y(1) &= A \sum_{n=0}^{N-1} T_c(1,n)x(n) \\ 5 \quad y(2) &= A \sum_{n=0}^{N-1} T_c(2,n)x(n) \\ &\vdots \\ y(N-1) &= A \sum_{n=0}^{N-1} T_c(N-1,n)x(n) \end{aligned}$$

根據上述之展開式，顯示在利用轉換函數將輸入訊號  $x(n)$  轉換為輸出訊號  $y(k)$  的過程中，需要運用到乘法運算、累加運算、及乘以常數等三部分，由於本發明係藉由固定數值乘法及選擇分配來取代習知之乘法運算，因此形成如  
10 圖2所示之硬體架構設計圖。其中，固定數值乘法器12是將輸入訊號  $x(n)$  乘以所有的轉換係數，再經由路徑選擇器(mux)13根據轉換函數之定義而將相乘結果分配到累加器141,142,143，因此路徑選擇器13還需要一個控制器131來產生控制訊號；累加器141,142,143則是將自路徑選擇器13  
15 傳來之數值進行累加運算；最後分別由乘法器151,152,153將累加結果乘以一個固定常數項  $A$ ，再透過輸出單元16將計算出的輸出訊號  $y(k)$  加以輸出。

第一實施例：

請參閱圖3之流程圖，第一實施例將以四點傅立葉轉換為基礎，詳述本發明將習知轉換函數之硬體架構設計為如圖2所示之硬體架構之程序。

於本實施例中，輸入單元11所擷取之轉換函數(步驟5 S301)係以下列矩陣型態表示：

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = T_x = \begin{bmatrix} e^{-j\frac{0\pi}{4}} & e^{-j\frac{0\pi}{4}} & e^{-j\frac{0\pi}{4}} & e^{-j\frac{0\pi}{4}} \\ e^{-j\frac{0\pi}{4}} & e^{-j\frac{2\pi}{4}} & e^{-j\frac{4\pi}{4}} & e^{-j\frac{6\pi}{4}} \\ e^{-j\frac{0\pi}{4}} & e^{-j\frac{4\pi}{4}} & e^{-j\frac{8\pi}{4}} & e^{-j\frac{12\pi}{4}} \\ e^{-j\frac{0\pi}{4}} & e^{-j\frac{6\pi}{4}} & e^{-j\frac{12\pi}{4}} & e^{-j\frac{18\pi}{4}} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}。$$

由於本實施例之轉換函數中，有部分轉換係數值是相同的，因此可將具有相同數值之轉換係數根據下列公式加以簡化(步驟S302)、進而統一為數值一致之轉換係數：

$$e^{j(0+2l\pi)} = e^{j0} \quad l \in \text{整數}，$$

並且形成如下所示之矩陣：

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} e^{-j\frac{0\pi}{4}} & e^{-j\frac{0\pi}{4}} & e^{-j\frac{0\pi}{4}} & e^{-j\frac{0\pi}{4}} \\ e^{-j\frac{0\pi}{4}} & e^{-j\frac{2\pi}{4}} & e^{-j\frac{4\pi}{4}} & e^{-j\frac{6\pi}{4}} \\ e^{-j\frac{0\pi}{4}} & e^{-j\frac{4\pi}{4}} & e^{-j\frac{0\pi}{4}} & e^{-j\frac{4\pi}{4}} \\ e^{-j\frac{0\pi}{4}} & e^{-j\frac{6\pi}{4}} & e^{-j\frac{4\pi}{4}} & e^{-j\frac{2\pi}{4}} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}。$$

接著係使用固定數值乘法器12取代傳統乘法器以實現乘法運算，而形成如圖4所示之初步硬體架構。需注意的是，傳統乘法器亦用以處理轉換係數與輸入訊號之相乘運算，惟在習知硬體架構中，乘法器所接收之轉換係數值會隨著時序不同而改變，故並非進行『固定數值』乘法運算，其必須藉由記憶體來儲存係數值、再依據時序讀入對應係

數，過程相當複雜繁瑣且耗用系統資源；而固定數值乘法器12之特色即在於每一固定數值乘法器12僅需處理一特定係數值與輸入訊號之乘法運算，俾能簡化運算過程。

圖4之硬體架構係根據不同時序( $n=0,1,2,3$ )分別建構出固定數值乘法器，但實際上在固定數值乘法運算的過程中，僅使用到 $e^{-j\frac{0\pi}{4}}$ 、 $e^{-j\frac{2\pi}{4}}$ 、 $e^{-j\frac{4\pi}{4}}$ 、及 $e^{-j\frac{6\pi}{4}}$ 四個轉換係數，因此可將使用相同轉換係數之固定數值乘法器12合併集中在一起而形成如圖5所示之硬體架構(步驟S303)，以避免使用多餘的固定數值乘法器12進行重複乘法運算之情形。

又，由於大部分轉換函數中之轉換係數都具有對稱關係，因此可利用此性質來簡化固定數值乘法器12的數量(步驟S304)。於本實施例中， $e^{-j\frac{2\pi}{4}}$ 、 $e^{-j\frac{4\pi}{4}}$ 、及 $e^{-j\frac{6\pi}{4}}$ 分別與 $e^{-j\frac{0\pi}{4}}$ 相差 $-j$ 、 $-1$ 、及 $j$ 倍，使得固定數值乘法器12可簡化為如圖6所示，顯示固定數值乘法器12的數值只剩下原本的四分之一(僅剩一共用之固定數值乘法器12)，足證利用轉換係數間的對稱關係將可大幅減少硬體架構。此外，由於 $f_0$ 及 $f_1$ 分別與 $f_2$ 及 $f_3$ 差 $-1$ 倍，所以硬體可以先進行 $f_0$ 及 $f_1$ 的運算，再經由路徑選擇器13的控制器131將 $f_2$ 及 $f_3$ 計算出來，以節省路徑選擇器13的複雜度。

除了利用轉換係數間之關係來集中處理乘法以簡化硬體架構之外，由於本實施例係採用固定數值乘法運算，因此僅利用加法器與減法器就可以實現乘法器的功能。當輸入訊號乘以一固定數值(即轉換係數)時，可表示為：

$$G = Dx(n) ,$$

其中，D為轉換係數，並可以二進制表示為：

$$D = \sum_{i=0}^{L-1} d_i 2^i ,$$

其中， $d_i$ 為0或1，L代表轉換係數的位元長度，並可據此將  
5 輸入訊號乘以轉換係數之公式改寫為：

$$G = \sum_{i=0}^{L-1} d_i x(n) 2^i .$$

由於 $d_i$ 只有0或1的可能，所以 $x(n)$ 乘以 $d_i$ 後， $x(n)$ 不是  
等於0就是維持不變，而乘以 $2^i$ 則相當於位元位移，所以上  
述方程式只需要加法器就可完成。例如以十進制表示之轉  
10 換係數  $D_1 = 0.61676025390625_{(10)}$  可用二進制表示如下：

$$D_1 = 0.10011101111001_{(2)} ,$$

並將轉換係數 $D_1$ 代入轉換函數後計算出輸出訊號為：

$$G = (x(n) \gg 1) + (x(n) \gg 4) + (x(n) \gg 5) + (x(n) \gg 6) + (x(n) \gg 8) \\ + (x(n) \gg 9) + (x(n) \gg 10) + (x(n) \gg 11) + (x(n) \gg 14) .$$

15 請一併參閱圖7，顯示輸入訊號 $x(n)$ 乘以轉換係數 $D_1$   
之固定數值乘法運算僅透過八個加法器即可完成。

當固定數值相乘運算要用加法器實現時，其加法的數  
目將取決於固定數值以二進制表示時『1』的數目，所以當  
『1』的個數越少，加法器的個數就會越少。因此可採用典  
20 型有號位元(Canonic Singed Digit, CSD)表示法以減少『1』  
的個數，其係將位元數值表示為具有-1、0、1三種表示法，  
並使用一個『1』及『-1』來取代連續的『1』，例如『15』  
係以二進制表示為『1111』，但由於 $15 = 16 - 1$ ，故可將『16-1』

用CSD表示為 $1000\bar{1}$ ，使得原本非0的個數由四個簡化為二個。同理，轉換係數 $D_1$ 亦可使用CSD法表示為：

$$D_1 = 0.101000\bar{1}000\bar{1}001_{\text{CSD}},$$

因此，輸出訊號可改寫為：

$$5 \quad G = (x(n) \gg 1) + (x(n) \gg 3) - (x(n) \gg 7) - (x(n) \gg 11) + (x(n) \gg 14)。$$

請參閱圖8，顯示於本實施例中，以CSD法表示之轉換係數 $D_1$ 僅需四個加減法器即可實現相同的固定數值乘法運算結果，優於以二進制表示法所需使用的八個加法器。

此外，由於固定數值乘法器12是將輸入訊號乘以同一個固定數值，因此當固定數值越多或表示的位元越長時，  
10 硬體架構的重複性就會越高，故本實施例可在擷取出所有轉換係數之數值後(步驟S305)，找出轉換係數間之共享子項(shared term)以對固定數值乘法器12再進行簡化(步驟S306)。例如轉換函數具有二轉換係數 $D_1 = 0.61676025390625$

15 及 $D_2 = 0.28753662109375$ ，其二進制表示法分別為：

$$D_1 = 0.10011101111001_{(2)},$$

$$D_2 = 0.01001001100111_{(2)},$$

其中，轉換係數 $D_1$ 及 $D_2$ 同時具有『1001』、『11』、及『111』之架構，亦即轉換係數 $D_1$ 及 $D_2$ 可共用此三個相同的架構(即  
20 共享子項)，藉由共享子項將建構出如圖9所示之硬體架構，係由八個加法器所組成，其中，A為『1001』，B為『11』，C為『111』。需注意的是，若未在轉換係數 $D_1$ 及 $D_2$ 間擷取出共享子項，則轉換係數 $D_1$ 需使用八個加法器(因為轉換係



數  $D_1$  中有九個『1』，故需使用八個加法器方能完成固定數值乘法)、轉換係數  $D_2$  則需要使用六個加法器，總計十四個加法器，顯示利用共享子項有助於減少加法器的使用個數。

5 同理，若將轉換係數  $D_1$  及  $D_2$  以 CSD 法表示為：

$$D_1 = 0.101000\bar{1}000\bar{1}001_{\text{CSD}},$$

$$D_2 = 0.01001010\bar{1}0100\bar{1}_{\text{CSD}},$$

則可自其中擷取出共享子項『101』及『 $\bar{1}001$ 』，並建構出如圖 10 所示之硬體架構，係由七個加法器組成，其中，D 為『101』，E 為『 $\bar{1}001$ 』。同理，若未擷取共享子項，則以 CSD 法表示之轉換係數  $D_1$  及  $D_2$  總計將使用九個加法器，亦大於利用共享子項所需的七個加法器個數。

轉換係數除了以二進制或 CSD 法表示之外，亦可採用其他表示法，例如混合式有號位元 (Hybrid Signed Digit, HSD) 表示法，其對每個位元都可以有號位元 (signed digit) 或無號位元 (unsigned digit) 表示，亦如為有號位元時可用 -1、0、及 1 來表示，無號位元則將此位元用 0 和 1 表示，因此轉換係數  $D_1$  及  $D_2$  可用 HSD 法表示如下：

$$D_1 = 0.100 \frac{1}{1} 00\bar{1}000\bar{1}001_{\text{HSD}},$$

$$20 \quad D_2 = 0.010010100\bar{1}\bar{1}00\bar{1}_{\text{HSD}},$$

其共享子項為『1001』及『 $100\bar{1}$ 』，並可對應建構出如圖 11 所示之硬體架構，包括六個加法器，其中，F 為『1001』，H 為『 $100\bar{1}$ 』。根據上述之說明，顯示在把轉換函數的所有

相乘累加器之乘法器集中處理後，當各轉換函數間具有相同架構之共享子項時，將可減少加法器之個數；亦即當固定數值乘法器12中的數值越多時，就可以產生較多的共享子項，使每一個數值就越容易使用較少的共享子項來組合，進而減少固定數值乘法器12中平均每一數值所使用到之加法器個數。

而在使用由加減法器所組成之固定數值乘法器12完成相乘運算後，控制器13係產生控制訊號以透過路徑選擇器13將相乘結果分配至輸出訊號 $y(k)$ 之時序所對應的累加器141,142,143(步驟S307)；並在完成累加運算後(步驟S308)後藉由輸出單元16將輸出訊號 $y(k)$ 加以輸出(步驟S309)。其中，由於在本實施例四點離散傅立葉轉換之轉換函數中，輸入訊號 $x(n)$ 僅與轉換係數進行相乘運算，而無常數項A，故不需使用乘法器151,152,153來進行常數乘法運算(或將常數項視為『1』)，更可簡化轉換函數所需使用之硬體架構。

## 第二實施例：

本實施例係應用於離散多頻調變(Discrete Multi-Tone, DMT)系統，其在以離散多頻調變系統為基礎之非對稱數位用戶迴路(Asymmetrical Digital Subscriber Line, ADSL)中係使用512點反向離散傅立葉轉換(Inverse Discrete Fourier Transform, IDFT)運算作為調變，本實施例之轉換函數為：

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j \frac{2nk\pi}{N}} \quad \text{for } n = 0, 1, \dots, N-1,$$

其中，N為反向離散傅立葉轉換之點數(在非對稱數位用戶迴路中，N=512)，x(n)為時域上之輸出訊號，X(k)為頻域上之輸入訊號，且為了保持實數的時域訊號輸出，故頻域的輸入訊號必須與時域訊號呈共軛對稱，即如下所示之共軛對稱關係：

$$X(N-k) = X^*(k) \quad \text{for } k = 1, 2, \dots, \frac{N}{2}-1,$$

此外，非對稱數位用戶迴路中反向傅立葉轉換的輸入訊號的直流(DC)和奈氏頻率(Nyquist Frequency)成分需為零，即：

$$X(0) = X(N/2) = 0.$$

根據上述二式可將本實施例之轉換函數簡化改寫為：

$$x(n) = \frac{2}{N} \sum_{k=1}^{\frac{N}{2}-1} \Re \left\{ X(k) e^{j \frac{2nk\pi}{N}} \right\} \quad \text{for } n = 0, 1, \dots, N-1,$$

其中， $\Re \{ \}$ 為取實數部分。

請參閱圖2，於第二實施例之轉換函數中，乘係數及取實數部分 $\Re \{ \}$ 與輸入訊號間之相乘運算可透過圖2之固定數值乘法器12完成，之後同樣經由路徑選擇器13將相乘結果分配到適當的累加器141,142,142上以完成累加運算，而最後乘法器151,152,153的乘法係數A則相當於本實施例轉換函數之 $\frac{2}{N}$ ，其為2的次方項，故不需再加裝額外硬體架構。

以下將詳述本實施例固定數值乘法器12、路徑選擇器13與控制器131、累加器141,142,143、及乘法器151,152,153之設計原理。

本實施例之轉換係數  $e^{j\frac{2nk\pi}{N}}$  亦可如同第一實施例使用下列公式簡化為  $e^{j\frac{2\phi\pi}{N}}$  :

$$e^{j(\theta+2l\pi)} = e^{j\theta} \quad l \in \text{整數},$$

其中， $\phi = nk \% N$ ，即  $nk$  除以  $N$  取餘數，請參閱圖12，當轉換係數以單位圓表示時， $\phi = \frac{N}{2}$  代表相角為  $\pi$ ， $\phi = N$  相當於  $\phi = 0$ ，故可求出  $\phi$  範圍係從0到  $N-1$ ，總共具有512個數值。

此外， $x(n)$  與  $x\left(n + \frac{N}{2}\right)$  間之對應關係可根據轉換函數而計算出下列結果：

$$\begin{aligned} x(n) &= \frac{2}{N} \sum_{k=0}^{\frac{N}{2}-1} \Re \left\{ X(k) e^{j\frac{2nk\pi}{N}} \right\} \\ x\left(n + \frac{N}{2}\right) &= \frac{2}{N} \sum_{k=0}^{\frac{N}{2}-1} \Re \left\{ X(k) e^{j\frac{2nk\pi}{N}} \right\} e^{jk\pi} \end{aligned} \quad \text{for } n = 0, 1, \dots, \frac{N}{2} - 1.$$

由上述比較結果可知， $x(n)$  與  $x\left(n + \frac{N}{2}\right)$  相差  $e^{jk\pi}$  倍， $k$  為整數，顯示第  $n + \frac{N}{2}$  個輸出訊號將會與第  $n$  個輸出訊號相等或相差一負號。於本實施例中，固定數值乘法器12之轉換係數為單位圓上相角0到  $\pi$  之間的數值，亦即乘以  $e^{j\frac{2\phi\pi}{N}}$ ，而  $\phi$  從0到  $\frac{N}{2} - 1$ ，且只取實數項。由於第  $n$  個與第  $n + \frac{N}{2}$  個累加器

係從路徑選擇器13接收到同一訊號，因此必須由控制器131送出控制訊號來控制累加器是否需先乘以-1後再進行累加運算。如此一來，本實施例可據以將路徑選擇器13之硬體架構由原先的512輸入對512輸出簡化為256輸入對256輸出，而大幅降低分配複雜度。

接著，將複數的乘法展開後可計算出：

$$f_{\phi} = \Re \left\{ X(k) e^{j \frac{2\phi\pi}{N}} \right\} = X_r(k) \cos \frac{2\phi\pi}{N} - X_i(k) \sin \frac{2\phi\pi}{N} \quad \text{for } \phi = 0, 1, \dots, \frac{N}{2} - 1,$$

其中， $X_r(k)$ 與 $X_i(k)$ 分別為輸入訊號的實部及虛部，因此請參閱圖13之硬體架構，固定數值乘法器12將先把轉換係數分為兩個實數運算後，再使用減法器進行相減運算，其中 $F(x)$ 為餘弦的固定數值乘法運算， $F''(x)$ 則為正弦的固定數值乘法運算。

於 $F(x)$ 中，由於固定係數為0到 $\pi$ 之間的餘弦值，因此利用三角函數的餘弦的對稱關係 $\cos(\theta) = -\cos(\pi - \theta)$ ，可將 $F(x)$ 簡化為：

$$\begin{aligned} f_{\phi} &= X_r(k) \cos \frac{2\phi\pi}{N} \quad \text{for } \phi = 0, 1, \dots, 127, \\ f_{\phi} &= -f_{256-\phi} \quad \text{for } \phi = 129, 130, \dots, 255 \end{aligned}$$

使得餘弦的係數項減少一半，進而將 $F(x)$ 之硬體架構簡化為如圖14所示，且當 $\phi = \frac{N}{4}$ 時，係數值為0可忽略不計，其中

$P(x)$ 係用以進行0到 $\frac{N}{4} - 1$ 的乘法運算，其如下所示：

$$f_{\phi}' = X_i(k) \cos \frac{2\phi\pi}{N} \quad \text{for } \phi = 0, 1, \dots, \frac{N}{4} - 1。$$

同理，在  $F''(x)$  中，正弦函數係於 0 到  $\pi$  之間對  $\frac{\pi}{2}$  對稱，

亦即當角度為  $\frac{2\phi\pi}{N}$  時，相當於角度等於  $\pi - \frac{2\phi\pi}{N}$  的正弦值，而

將  $F''(x)$  簡化為：

$$f_{\phi}'' = X_i(k) \sin \left( \frac{2\phi\pi}{N} \right) \quad \text{for } \phi = 1, 2, \dots, 128。$$

$$f_{\phi}'' = f_{256-\phi}'' \quad \text{for } \phi = 129, 130, \dots, 255$$

同樣亦可使正弦的係數項減少一半，而將  $F''(x)$  之硬體架構簡化為如圖 15 所示，且當  $\phi=0$  時，正弦的係數值為 0 可忽略不計，其中  $P'(x)$  則用以進行下列運算：

$$f_{\phi}'' = X_i(k) \sin \frac{2\phi\pi}{N} \quad \text{for } \phi = 1, 2, \dots, \frac{N}{4}。$$

本實施例轉換函數之硬體架構在進行簡化之前，係定義有  $N$  個複數乘法，若利用只取實數項輸出、則固定數值乘法器 12 總共會有  $2N$  個固定數值。而在根據轉換係數間的對稱關係進行簡化後，本實施例僅需針對  $P(x)$  及  $P'(x)$  來設計硬體架構(即圖 14 及圖 15 之硬體架構)，其分別只有  $\frac{N}{4}$  個

實數乘法，所以總共只會用到  $\frac{N}{2}$  個固定數值，比起簡化前需使用  $2N$  個固定數值之架構節省了約四倍，同時也將路徑選擇器 13 的硬體架構節省了一半(由 512 對 512 簡化為 256 對 256)。

接著，還可利用共享子項及加減法器的組合來對 $P(x)$ 及 $P'(x)$ 進行簡化。由於擷取共享子項之原理與第一實施例相同，故不在此贅述。需注意的是，由於 $\sin(\theta) = \cos\left(\frac{\pi}{2} - \theta\right)$ ，因此可將正弦函數 $P'(x)$ 改寫為：

$$5 \quad f_{\phi}'' = X_i(k) \cos\left(\frac{2(N/4 - \phi)\pi}{N}\right) \quad \text{for } \phi = 1, 2, \dots, \frac{N}{4},$$

如此一來，將可使用相同架構來實現 $P'(x)$ ，而不需另外定義原先 $P'(x)$ 的係數表示法，因此可據以建構出如圖16所示之固定數值乘法器12的硬體架構。其中，雖然 $P(x)$ 與 $P'(x)$ 已改寫為具有相同架構，但輸入訊號仍不相同，且 $f_0''$ 的輸出端為0，故 $f_0 = f_0'$ ，而 $f_{128}'$ 的輸出端為0，故 $f_{128} = -f_{128}''$ 。

由於路徑選擇器13必須將固定數值乘法器12的結果正確地分配到各個累加器141,142,143，且每個累加部分會依時序的不同分別對訊號 $X(k)e^{j\frac{2\phi\pi}{N}}$ 進行累加運算， $\phi$ 為0到 $N-1$ 。而根據先前所述，路徑選擇器13只會傳送 $\phi$ 從0到 $\frac{N}{2}$ 的訊號到累加器141,142,143上，也就是圖12單位圓上角度從0到 $\pi$ 的值，因此當累加器需接到 $\phi$ 從 $\frac{N}{2}$ 到 $N-1$ 的訊號時，只需要再乘以-1即可。故本實施例之路徑選擇器13與輸入/輸出訊號之關係即可以下列關係式表示：

$$S_n = f_{\psi} \quad \psi = \phi \% 2 = (nk) \% (N/2)。$$

20 本實施例需設計出一256對256之路徑選擇器13。首先可以2對2的路徑選擇器為例，其架構如圖17所示，顯示有

兩條控制訊號  $C_0$  及  $C_1$ ， $C_0$  係控制  $B_0$  的選擇、 $C_1$  則是控制  $B_1$  的選擇，且當控制訊號為 0 時選擇  $A_0$ ，反之選擇  $A_1$ 。接著可進一步利用 2 對 2 路徑選擇器來設計如圖 18 所示之 4 對 4 路徑選擇器，其中， $B_n$  的控制訊號係定義為：

$$5 \quad C_n(1,0) = \sum_{i=0}^1 C_n(i)2^i,$$

當中， $n$  由 0 到 3，例如  $B_n$  欲選擇  $A_2$  的訊號，而 2 的二進制表示法為『 $10_{(2)}$ 』，故  $C_n(0)$  為 0、 $C_n(1)$  為 1。但  $B_n$  的最低位元控制訊號可能被另一個輸出端的控制訊號控制，例如  $B_3$  欲選擇  $A_1$  時，控制訊號  $C_3(0)$  等於 1， $C_3(1)$  等於 0，則 MUX-2(4) 10 將會選擇上述路徑，亦即連結到 MUX-2(1)，其控制訊號為  $C_1(0)$  而非  $C_3(0)$ ，因此，圖 18 的架構必須在控制訊號  $C_n(0)$  與  $C_{n+2}(0)$  相同時才可正常的處理，不過在此路徑選擇器中，輸出端  $B_n$  的控制訊號為  $n$  乘以  $k$  取最低的兩個位元， $k$  為常數，則  $B_n$  控制訊號的最低位元  $C_n(0)$  將表示如下：

$$15 \quad C_n(0) = (nk) \% 2,$$

且控制訊號  $B_{n+2}$  的最低位元  $C_{n+2}(0)$  將可表示為：

$$\begin{aligned} C_{n+2}(0) &= ((n+2)k) \% 2 \\ &= (nk + 2k) \% 2 \\ &= ((nk) \% 2 + (2k) \% 2) \% 2 \\ &= ((nk) \% 2 + 0) \% 2 \\ &= (nk) \% 2 \end{aligned}$$

同理，可將上述之路徑選擇器逐漸擴充為適用於本實施例之 256 對 256 之路徑選擇器 13，其係為  $n$  乘以  $k$  取其第 0 20 位元到第 7 位元的數值，當  $n$  為 2 的倍數時，可以將其他  $n$  乘



以k的數值利用位移方式產生；而當n不為2的倍數時，則可利用其他結果的組合來產生，例如當n等於5時，可表示為：

$$5k = (1 + 4)k = 1k + 4k,$$

故顯示僅需使用一個加法器即可，餘此類推，因此總計需要  
5 要使用127個加法器。此外，控制器131係用以產生控制訊號來控制路徑選擇器13動作，但實際上由控制器131所產生之訊號中，有些位元固定為0，例如當n等於0時，所產生的位元必定全部為0，而當n等於6時，則第0位元也必定為0，因此當n為2的倍數時，則在低的位元會固定為0，這些固定  
10 為0的位元所控制的MUX則會固定選擇一個輸入訊號，藉以簡化MUX的個數。

最後，在累加器141,142,143的設計上，請參閱圖19，由於累加器係對路徑選擇器分配的結果進行累加運算，而且需要控制訊號來判斷輸入數值是否應乘以-1，因此具有  
15 一互斥或(XOR)閘來控制輸入訊號是否要乘以-1，當 $A_n$ 為0則選擇原本的輸入訊號，反之則將輸入訊號乘以-1。

於本實施例中，若 $\phi \geq 256$ ，接收訊號與實際要累加訊號相差-1倍，若 $\phi < 256$ ，則接收訊號就是實際要進行累加運算的訊號。因此當 $\phi$ 以二進制表示時，可利用 $\phi$ 以二進制表  
20 示時的第8個位元來作為累加器是否需乘以-1的控制訊號。在前述控制器131的設計中，係將n乘以k所產生的值取第0到第7位元，由於此階段需取得 $\phi$ 的第8個位元，所以必須將控制器131改為取第0到第8位元，因此當n為一介於0

到255間的數值時，根據下列公式便可計算出 $\phi$ 第8個位元的數值：

$$A_n = C_n(8) \text{ for } n = 0, 1, \dots, 255,$$

5 若 $n$ 係介於256與511間，則可根據下列公式以計算出 $\phi$ 的第8個位元值：

$$A_n = C_{n-256}(8) \oplus k_0 \text{ for } n = 256, 257, \dots, 511.$$

$k_0$ 為時序信號的最低位元。根據上述之說明，顯示本發明所提出之轉換函數之硬體架構設計方法，係可利用由加減法器所組成之固定數值乘法器、及路徑選擇器來取代  
10 傳統的乘法器及記憶體；且本發明之設計方法係可根據轉換係數間之數值對稱性、架構共用性...等特性來簡化轉換係數乘法運算個數、及所需之加減法器個數，此外，當轉換係數所需表示的數值越少時，本發明之設計方法越可簡化硬體架構，尤其當應用於超大型積體電路設計(VLSI)中  
15 時，更可有效達到低硬體成本及高運算效率之目的，實為一大進步。

上述實施例僅係為了方便說明而舉例而已，本發明所主張之權利範圍自應以申請專利範圍所述為準，而非僅限於上述實施例。

20

## 五、圖式簡單說明

圖1係習知四點離散傅立葉轉換之硬體架構示意圖。

圖2係本發明轉換函數之硬體架構設計圖。

圖3係本發明第一實施例之流程圖。

圖4係本發明第一實施例使用固定數值乘法器取代乘法器所形成之初步硬體架構示意圖。

圖5係本發明第一實施例將固定數值乘法器合併集中後所形成之硬體架構示意圖。

5 圖6係本發明第一實施例依對稱性簡化轉換係數後所形成之固定數值乘法器示意圖。

圖7係本發明第一實施例以二進制表示轉換係數所形成之固定數值乘法器示意圖。

10 圖8係本發明第一實施例以CSD法表示轉換係數所形成之固定數值乘法器示意圖。

圖9係本發明第一實施例使用共享子項簡化以二進制表示之轉換函數後所形成之固定數值乘法器示意圖。

圖10係本發明第一實施例使用共享子項簡化以CSD法表示之轉換係數後所形成之固定數值乘法器示意圖。

15 圖11係本發明第一實施例使用共享子項簡化以HSD法表示之轉換係數後所形成之固定數值乘法器示意圖。

圖12係本發明第二實施例以單位圓表示512點IDFT之轉換係數之示意圖。

20 圖13係本發明第二實施例固定數值乘法器之內部架構示意圖。

圖14係本發明第二實施例 $F'(x)$ 之硬體架構示意圖。

圖15係本發明第二實施例 $F''(x)$ 之硬體架構示意圖。

圖16係本發明第二實施例改良後之固定數值乘法器之硬體架構示意圖。

圖 17 係 2 對 2 路徑選擇器之硬體架構示意圖。

圖 18 係 4 對 4 路徑選擇器之硬體架構示意圖。

圖 19 係本發明第二實施例累加器之硬體架構示意圖。

## 5 六、圖號說明

輸入單元 11

固定數值乘法器 12

路徑選擇器 13

控制器 131

累加器 141, 142, 143

乘法器 151, 152, 153

輸出單元 16

1. 一種轉換函數之硬體架構設計方法，主要包括下列步驟：

5 一轉換函數擷取步驟，係擷取一轉換函數，該轉換函數用以將一特定領域之輸入訊號 $x(n)$ 轉換為另一特定領域之輸出訊號 $y(k)$ ；

一依數值簡化轉換係數步驟，係將該轉換函數中具有相同數值之轉換係數簡化為一致之轉換係數，其中，每一特定數值之轉換係數係對應定義有一固定數值乘法器；

10 一乘法運算步驟，係分別使用該固定數值乘法器將該輸入訊號與對應之轉換係數進行相乘運算；

一分配步驟，係使用一路徑選擇器以根據該轉換函數之定義而將該輸入訊號與對應轉換函數之相乘結果分配至該輸出訊號之時序所對應之累加器；

15 一累加運算步驟，係使用該累加器將位於該時序之相乘結果進行累加運算；

一乘以常數步驟，係使用一乘法器將該累加結果乘以該轉換函數之常數項以計算出該輸出訊號；以及

一輸出步驟，用以輸出該輸出訊號。

20 2. 如申請專利範圍第1項所述之方法，其中，該轉換函數為 $y(k) = A \sum_{n=0}^{N-1} T_c(k,n)x(n)$   $k=0,1,2,\dots,N-1$ ，當中， $A$ 為常數項， $T_c(k,n)$ 為轉換係數值。

3.如申請專利範圍第2項所述之方法，其中，該轉換函數係用以進行反向離散傅立葉轉換，且A為 $\frac{1}{N}$ 。

4.如申請專利範圍第1項所述之方法，其中，於該簡化轉換函數步驟後更包括一依對稱性簡化轉換係數步驟，係將具有對稱性之轉換係數簡化為共用一固定數值乘法器。

5.如申請專利範圍第1項所述之方法，其中，該等轉換係數係以二進制表示。

6.如申請專利範圍第5項所述之方法，其中，每一轉換係數所對應定義之固定數值乘法器係由至少一加減法器所組成。

7.如申請專利範圍第6項所述之方法，其中，該乘法運算步驟係包括下列步驟：

擷取所有轉換係數之數值；

比對該等轉換係數以提取出具有相同架構之共享子項，每一共享子項係可由至少一加減法器計算出；以及

將該等轉換係數中具有該等共享子項之架構以該共享子項取代。

8.如申請專利範圍第7項所述之方法，其中，該等轉換係數係以典型有號位元表示。

9.如申請專利範圍第7項所述之方法，其中，該等轉換係數係以混合式有號位元表示。

10. 一種轉換函數之硬體裝置，主要包括：

一輸入單元，用以輸入一轉換函數，該轉換函數用以將一特定領域之輸入訊號 $x(n)$ 轉換為另一特定領域之輸出訊號 $y(k)$ ；

至少一固定數值乘法器，用以將該輸入訊號與該轉換函數中所定義之對應轉換係數進行相乘運算；

至少一路徑選擇器，係根據該轉換函數之定義以將該輸入訊號與對應轉換函數之相乘結果分配至該輸出訊號之時序所對應之累加器；

至少一累加器，每一累加器係對應於一特定時序，用以將位於該時序之相乘結果進行累加運算；

至少一乘法器，係將該累加結果乘以該轉換函數之常數項以計算出該輸出訊號；以及

一輸出單元，用以輸出該輸出訊號。

11. 如申請專利範圍第10項所述之裝置，其中，該轉換函數為 $y(k) = A \sum_{n=0}^{N-1} T_c(k,n)x(n)$   $k=0,1,2,\dots,N-1$ ，當中， $A$ 為常數項， $T_c(k,n)$ 為轉換後之係數值。

12. 如申請專利範圍第10項所述之裝置，其中，該等轉換係數係以二進制表示。

13. 如申請專利範圍第12項所述之裝置，其中，每一轉換係數所對應定義之固定數值乘法器係由至少一加減法器所組成。

14. 如申請專利範圍第10項所述之裝置，其中，該路徑選擇器係具有一控制器用以產生一控制訊號。

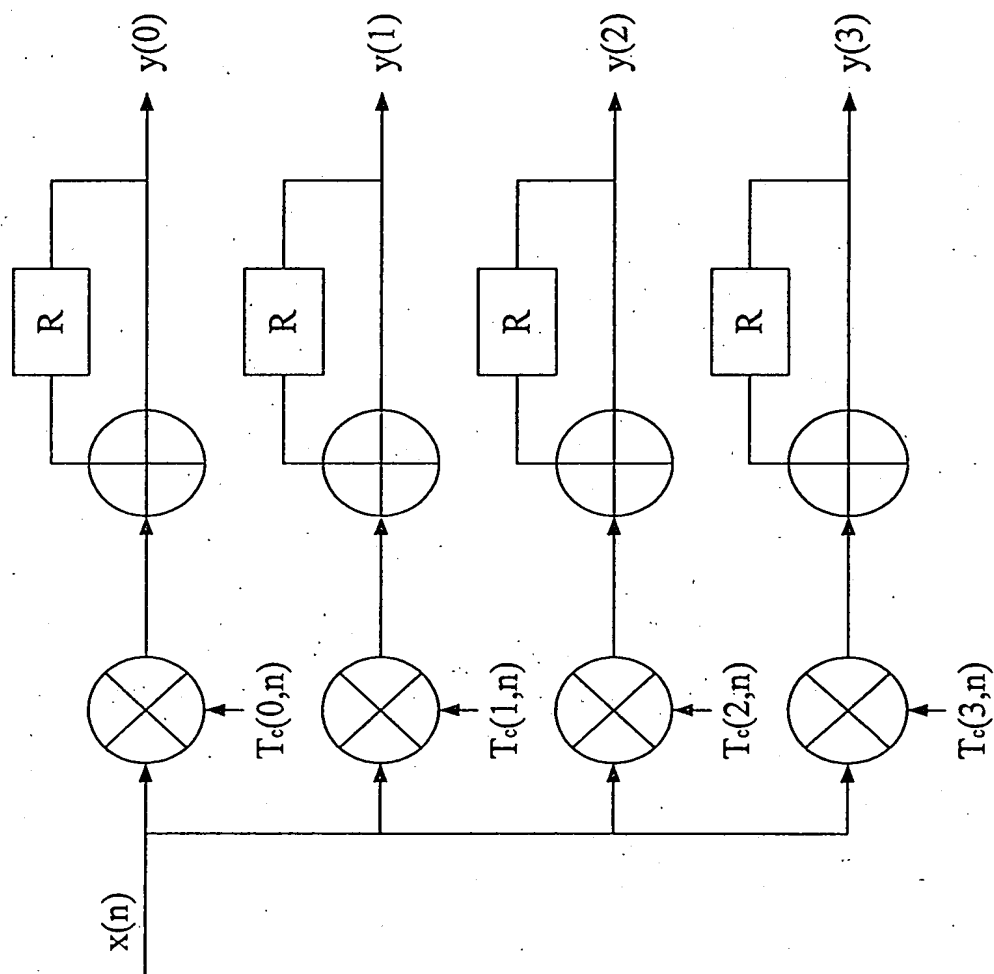


圖 1



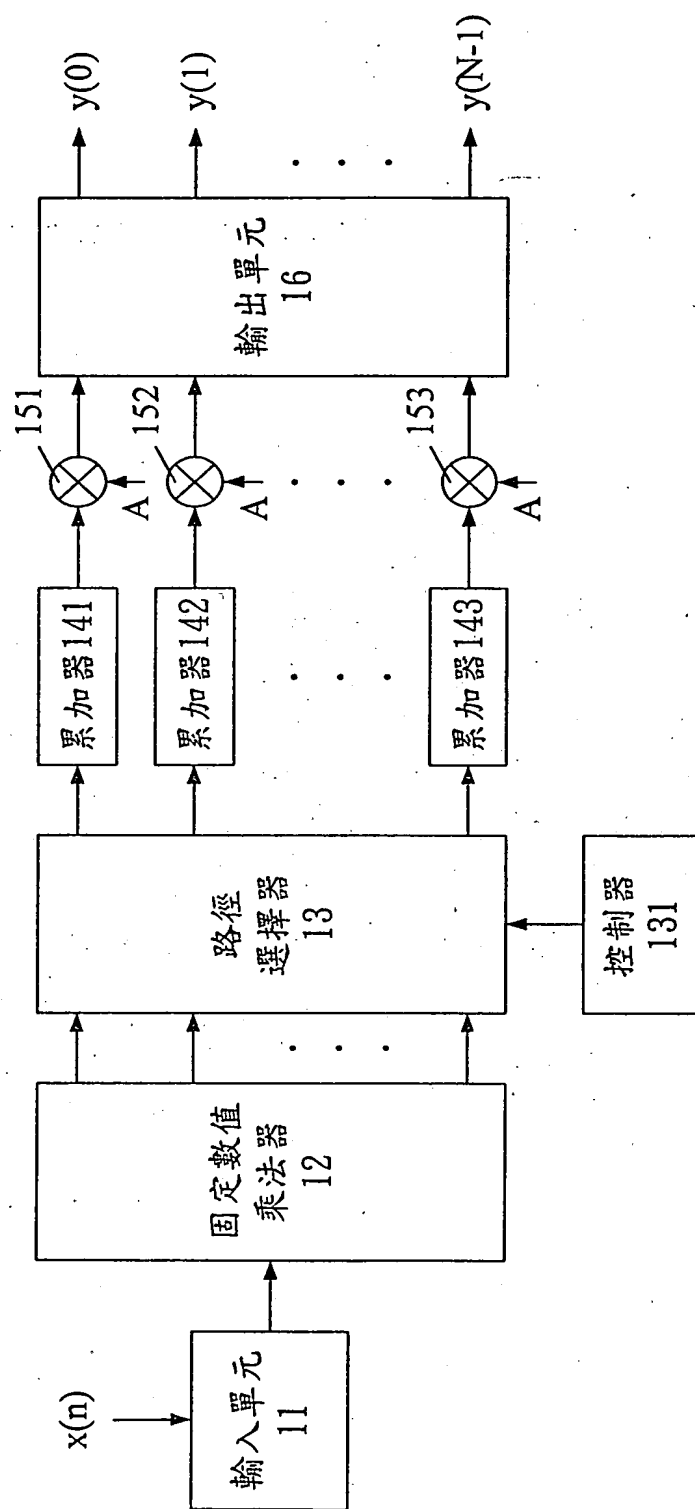


圖2

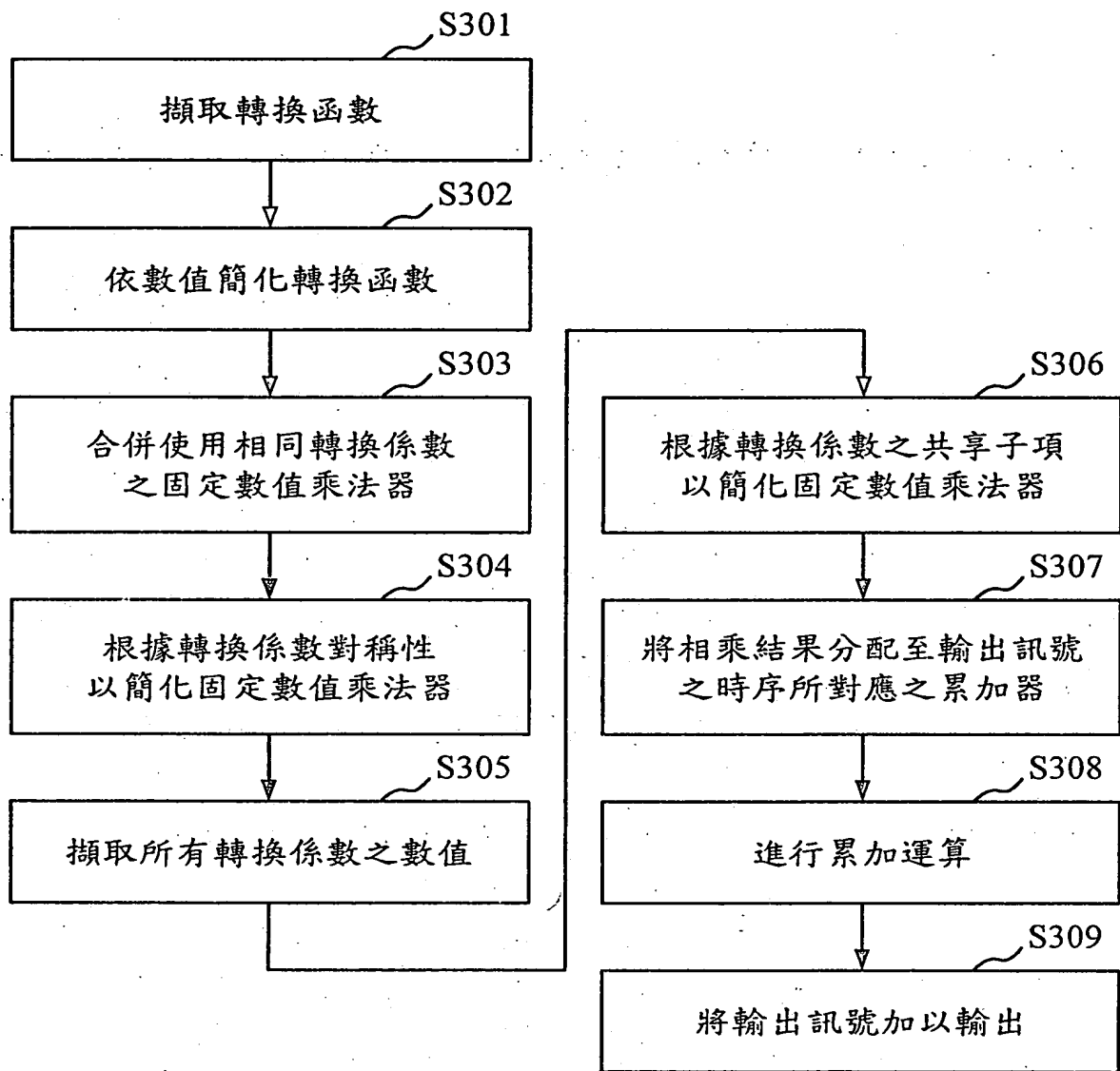


圖3

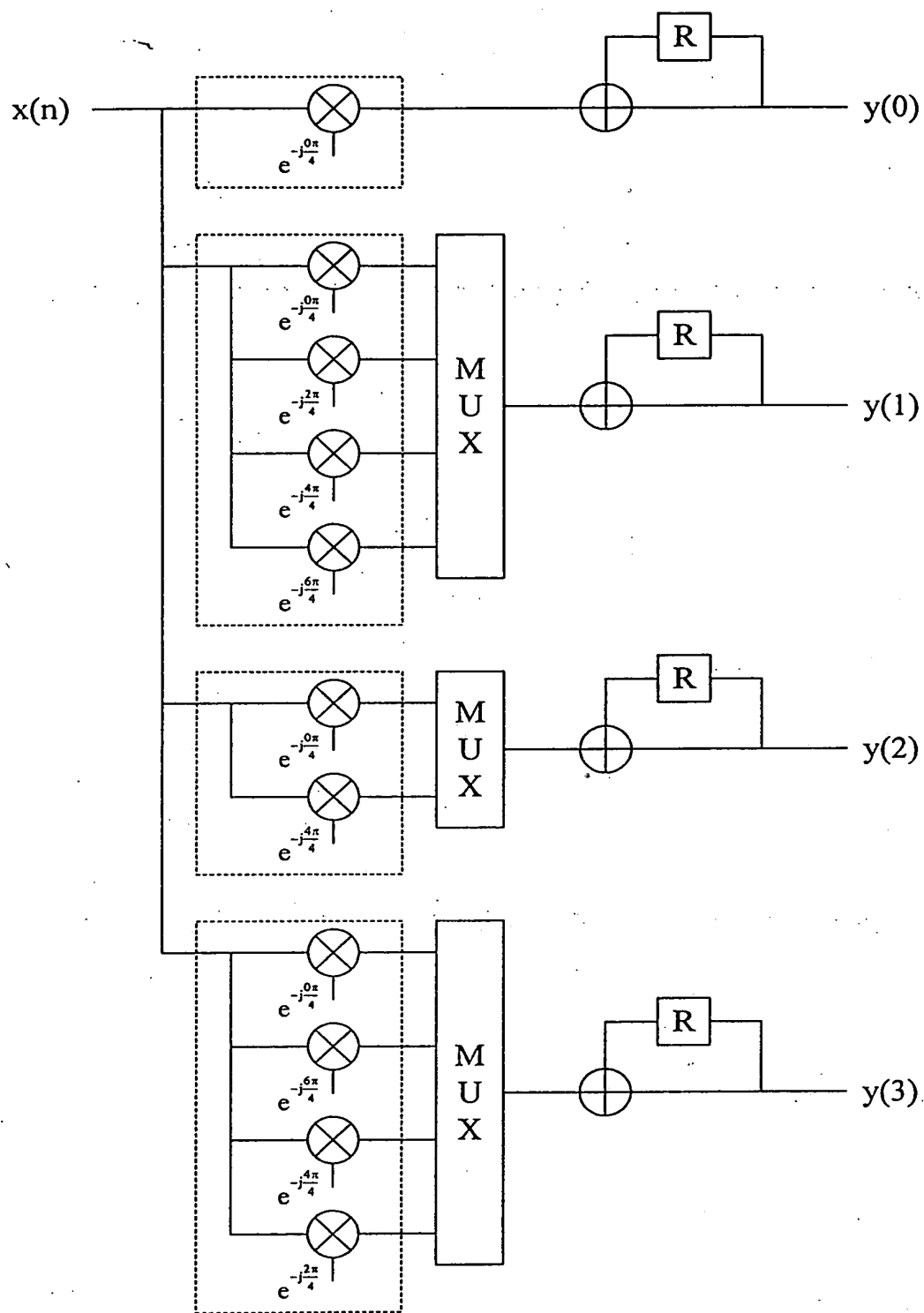


圖4

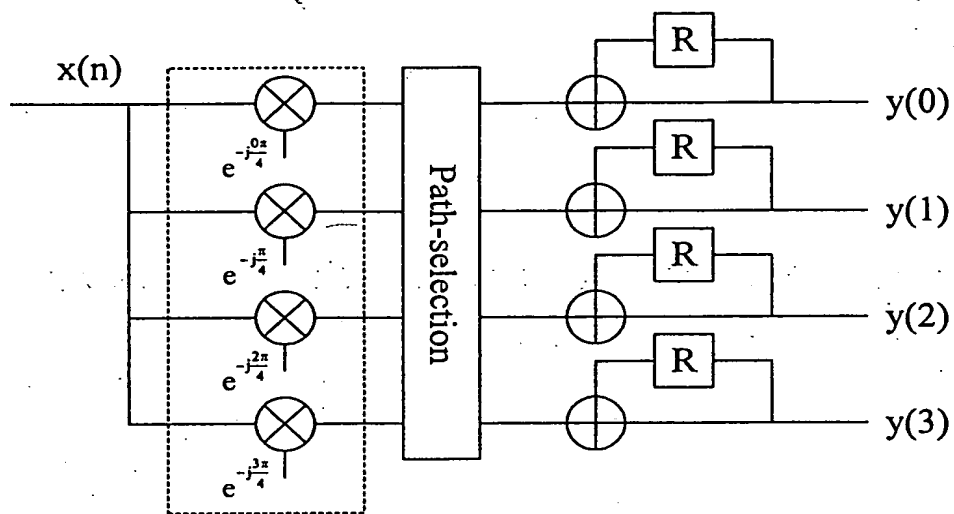


圖5

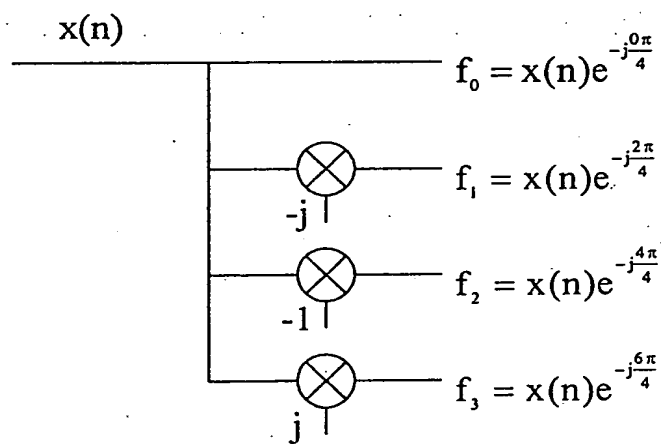


圖6

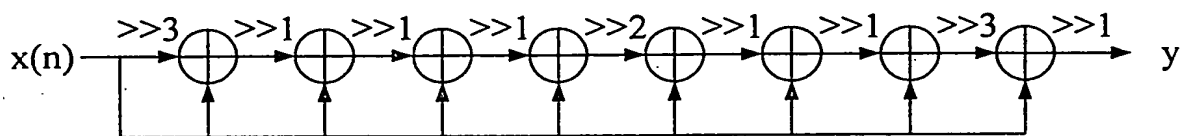


圖 7

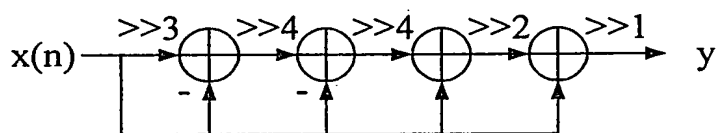


圖 8

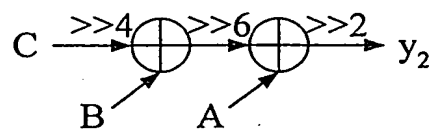
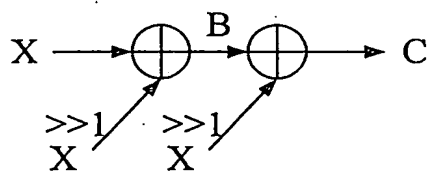
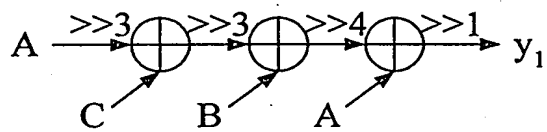
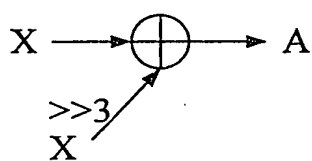


圖 9

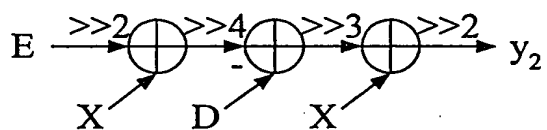
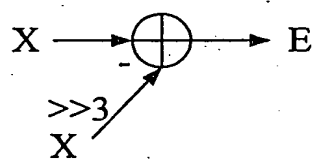
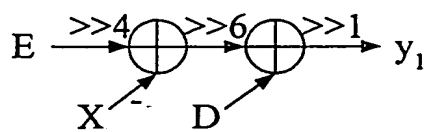
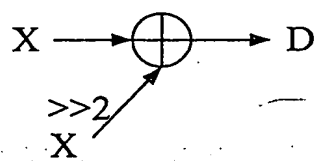


圖10

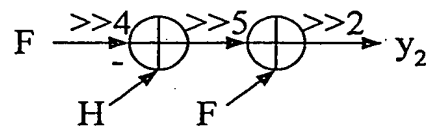
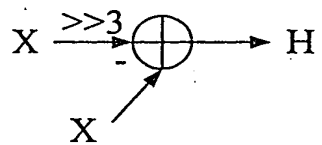
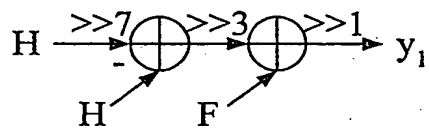
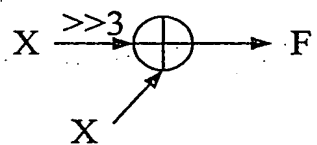


圖11

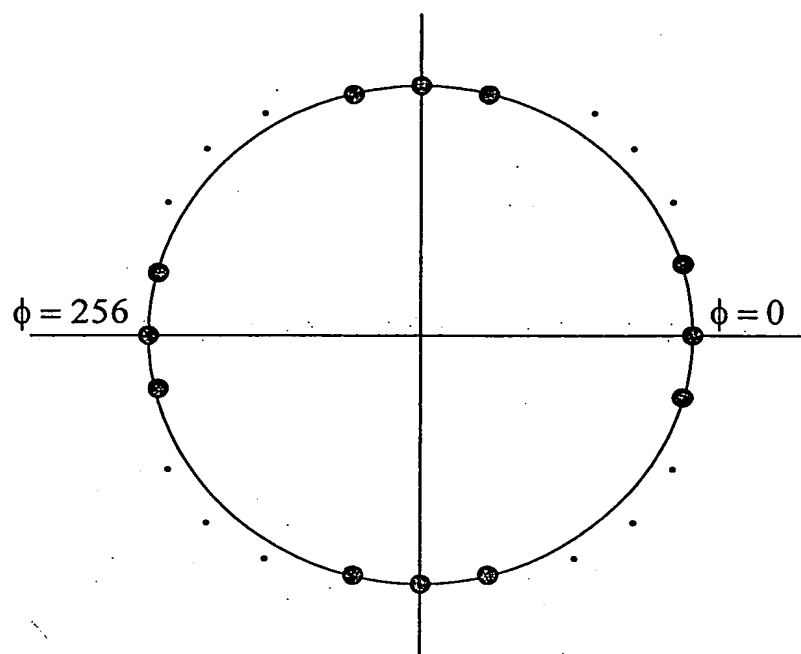


圖12

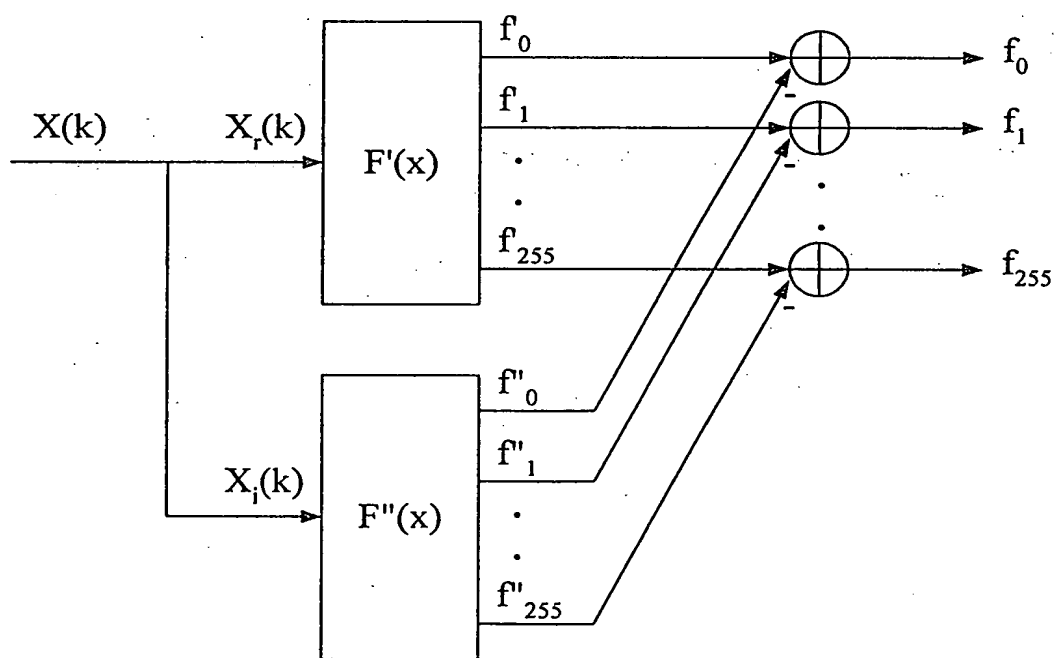


圖13

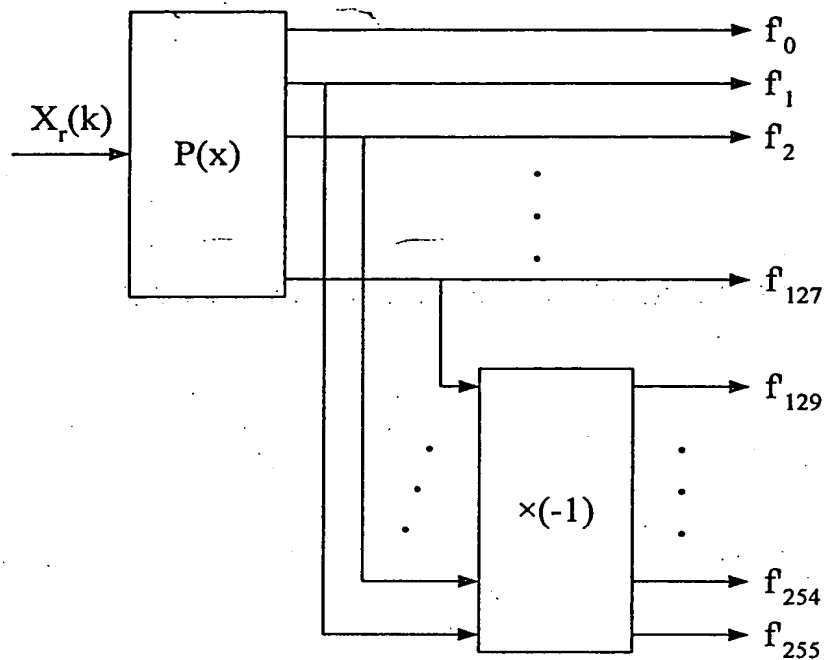


圖 14

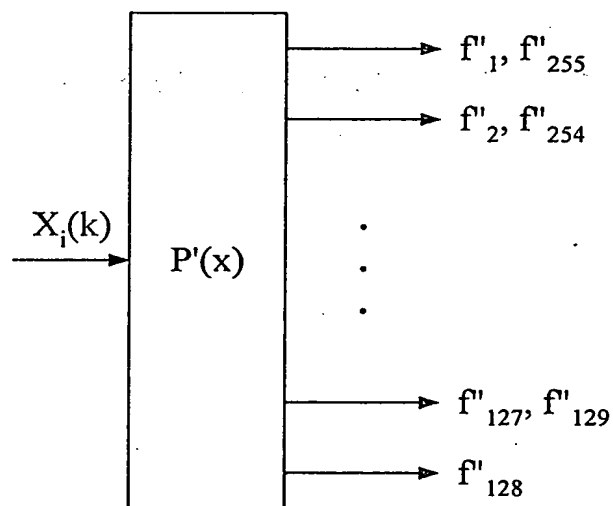


圖 15



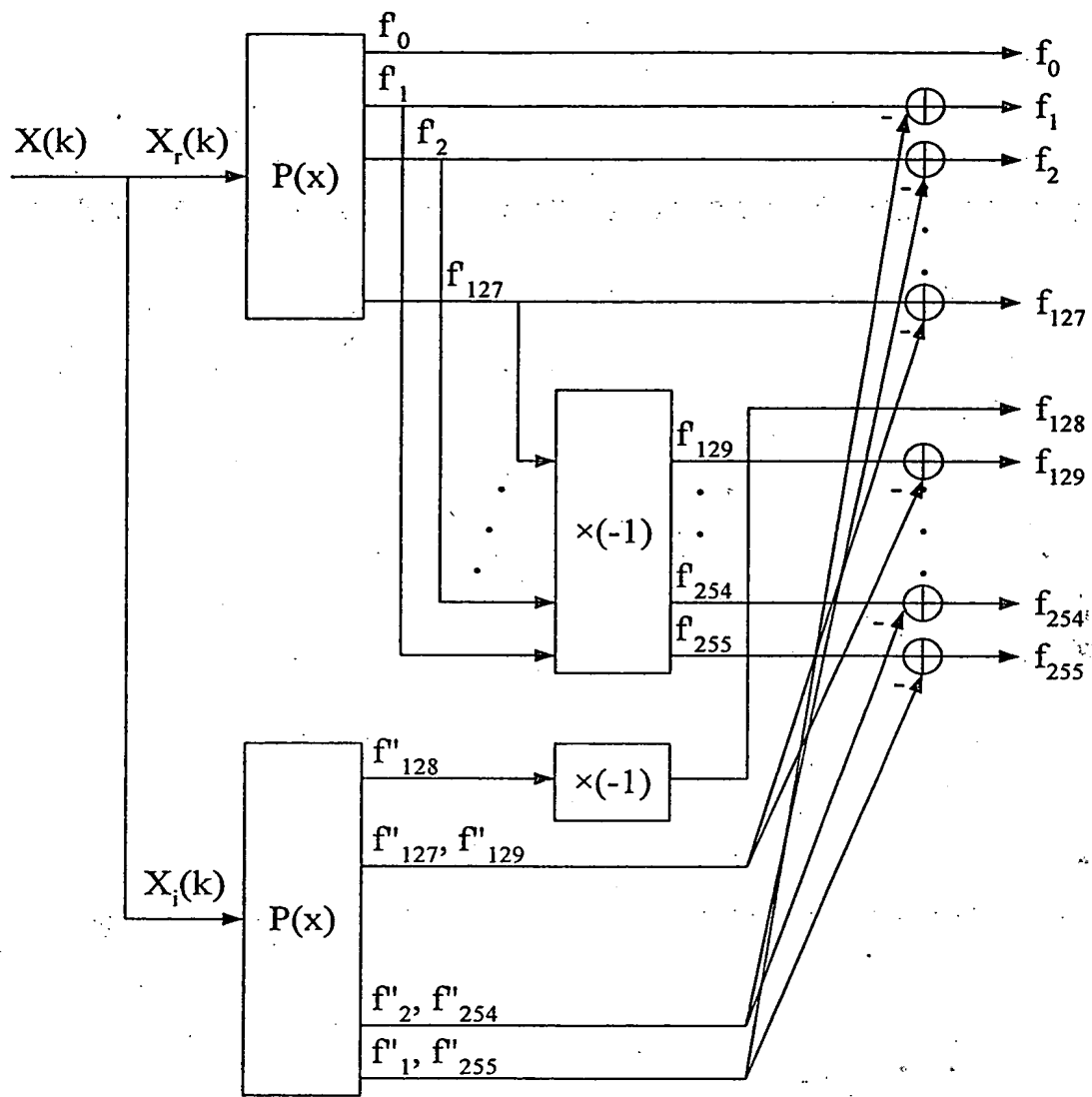


圖 16

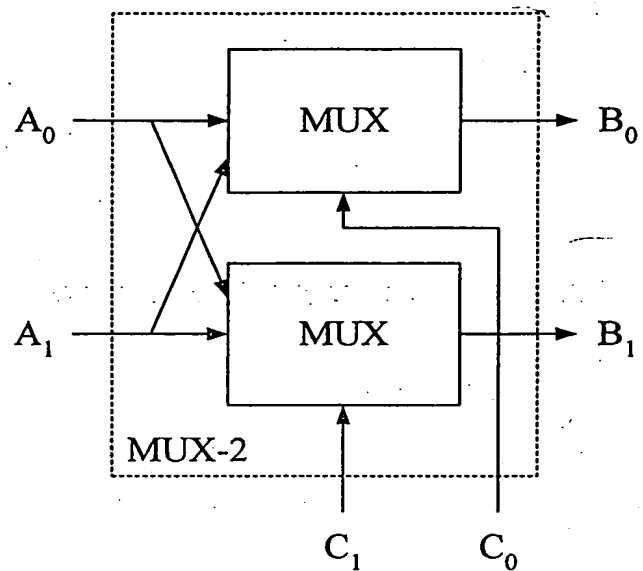


圖 17

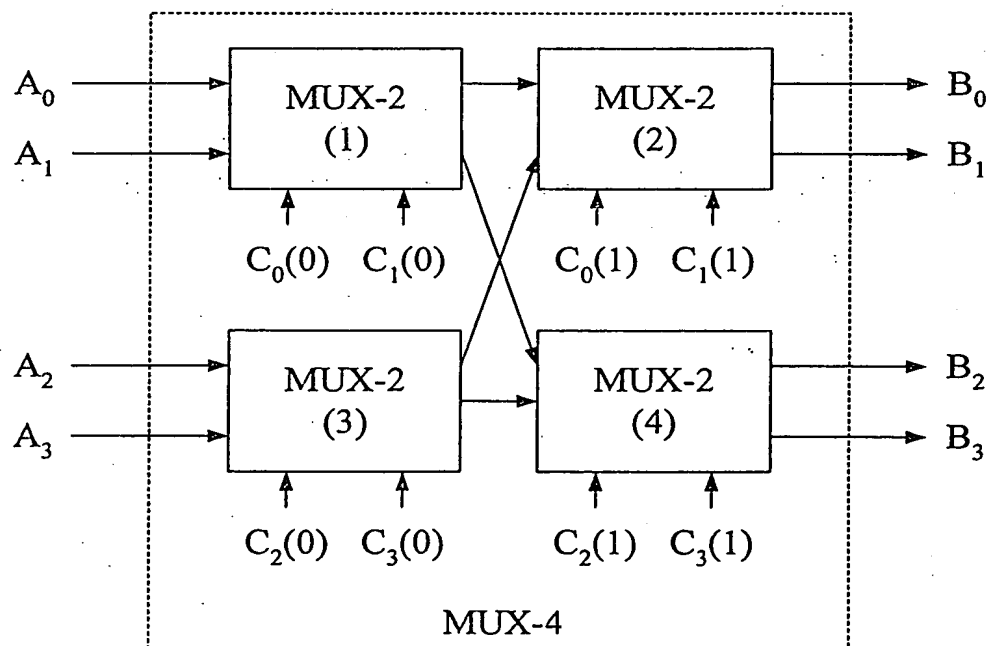


圖 18

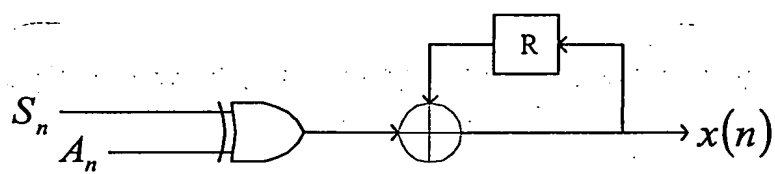


圖 19